Minicurso ministrado no SBIE 2007, 28 a 30 de Novembro de 2007, São Paulo, SP.

Chapter X

Objetos Digitais Complexos na Educação e os Objetos de Aprendizagem

André Santanchè, Andrea Lago, Patrícia Dourado Lima de Almeida, Pedro Ferreira Filho

Abstract

This text introduces the reader into the environment, which motivated the raising of the Learning Object (LO) notion, as well as the formation of standards aimed at sharing and reusing LOs. It will be presented not only the technical foundations of LOs, but also reflections on pedagogical challenges in view of conceiving and producing reusable content as objects. The text is illustrated with concrete examples that use tools for creation, edition, sharing and reuse of LOs, and their insertion in a Learning Management System – the Moodle. Besides the specific perspective in Education domain, it is concerned in a broader perspective of the Digital Complex Objects. The goal is to provide foundations for a reflection on the ways efforts related to LOs have followed, the purposes and strategies of which have many intersections with equivalent efforts in other domains.

Resumo

Este texto introduz o leitor no contexto que motivou o surgimento da noção de Objeto de Aprendizagem (OA), bem como a constituição de padrões para compartilhamento e reuso de OAs. Serão apresentados não apenas seus fundamentos técnicos, como também reflexões quanto aos desafios pedagógicos na concepção e produção de conteúdo reusável na forma de objetos. O texto é ilustrado com exemplos concretos através do uso de ferramentas para criação, edição, compartilhamento e reuso de OAs, bem como sua inserção em um Sistema para Gerenciamento de Cursos – o Moodle. Além da perspectiva mais específica no domínio da Educação, ele apresenta uma perspectiva mais ampla, a dos Objetos Digitais Complexos. O objetivo é fornecer subsídios para uma reflexão sobre os caminhos que seguem os esforços em torno dos OAs, cujos propósitos e estratégias têm grandes intercessões com esforços equivalentes em outros domínios.

Este texto é concedido sob uma Licença Creative Commons. Sob as seguintes condições: Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença. Para maiores informações veja licença completa no site creativecommons.org.

1. Introdução

A impressão causada pelo primeiro contato com a noção de Objeto de Aprendizagem (OA) traz a ambigüidade de uma idéia simultaneamente cativante e vaga. Cativa a idéia de poder criar estas "coisas", os objetos, os quais podem ser compartilhados e aplicados em atividades de ensino/aprendizagem. Objeto nos transmite a idéia de uma entidade que pode ser percebida e manipulada individualmente. Somos induzidos a imaginar repositórios, capazes de armazenar e classificar centenas desses objetos, de modo que possam posteriormente ser selecionados e reusados em contextos diversos.

Em contrapartida, uma reflexão mais aprofundada sobre o tema transmite a fragilidade de uma noção vaga. O que é exatamente um OA? Seria um software educacional completo, uma animação que é executada em um navegador Web, uma imagem? A tentativa de uma definição precisa comporta o risco dos extremos. De um lado, somos tentados a considerar tudo um OA e, neste caso, o termo perde o sentido. Como observam Sosteric and Hesemeier (2002), um termo que serve para definir tudo não define nada. Por outro lado, corremos o risco de criar definições extremamente restritivas, definindo um OA apenas como: um software executável, uma simulação, uma animação etc., e deixando de fora um rico conjunto de artefatos que podem igualmente ter um papel importante na construção de atividades de ensino-aprendizagem.

A caracterização de um OA deve considerar também sua natureza, que resulta da intersecção de dois domínios: Educação e Informática. Isto significa que não é suficiente definir um OA simplesmente do ponto de vista técnico da Informática, sem considerar os pressupostos pedagógicos da sua construção.

Um dos objetivos deste texto é dar forma à noção intuitiva dos OAs. Diante das diversas perspectivas que se formaram em torno desta noção intuitiva, não nos parece possível estabelecer uma definição única para o conceito. Por este motivo, trataremos do tema sob quatro perspectivas. Na Seção 2, traçaremos um cenário histórico, buscando precursores que certamente tiveram influência na formação da noção de objeto de aprendizagem, tais como os componentes educacionais. A Seção 3 é dedicada à compreensão do que é um OA; nela apresentaremos exemplos de OA e suas aplicações e levantaremos um conjunto de características que, a nosso ver, são imprescindíveis na compreensão de um OA. Na Seção 4, apresentaremos padrões abertos relacionados com Objetos de Aprendizagem. Na Seção 5, ampliaremos a nossa visão para um cenário maior, que envolve outros domínios além da Educação, e o objetivo será demonstrar que os OAs são parte de um movimento mais amplo, o dos Objetos Digitais Complexos. Finalmente, na Seção 6, sumarizamos as considerações finais e perspectivas futuras.

2. Precursores

Entre os meados e o final da década de 90, desponta um promissor campo de pesquisa, que combina a tecnologia dos componentes de software com a criação de software educacional. São os componentes de softwares educacionais (ou poderíamos chamá-los hoje em dia de componentes de aprendizagem).

Mesmo sendo considerado por alguns autores um termo pouco preciso [Olsen 2006], os componentes de software são uma tecnologia madura e a origem do termo data de 1968 [Mcilroy 1968]. Existem diferentes interpretações e implementações da noção de componente de software, mas elas sempre se baseiam em um conjunto fundamental de

características [Bachmann et al. 2000, Broy et al. 1998, Szyperski 2002]. Um componente, como indica o próprio termo, é uma unidade feita para compor algo. Componentes publicam a sua funcionalidade através de interfaces, que guiam a maneira como ele se relaciona com as demais entidades do sistema. As interfaces escondem detalhes de implementação, permitindo que o usuário abstraia detalhes internos do componente e se concentre na sua funcionalidade. Uma excelente metáfora apresentada em Roschelle et al. (1998) diz que poucas pessoas têm condições de montar o seu próprio aparelho estéreo combinando peças eletrônicas (transistores, resistores etc.), mas muitas conseguem fazê-lo comprando e ligando unidades maiores (tocador CD, caixa de som, amplificador etc.). Estas unidades maiores são os componentes.

Fundados sobre a tecnologia dos componentes de software, diversos projetos lançaram ferramentas nas quais os autores poderiam combinar peças para a construção de aplicações educacionais. Destacamos alguns destes projetos:

E-Slate
`http://e-slate.cti.gr/`
O E-Slate é um ambiente para a construção de micromundos por meio da combinação de componentes de software. A Figura 1 ilustra o processo de construção de micromundos. Inicialmente os componentes são inseridos no ambiente e configurados. A configuração é feita usando um painel no qual as propriedades do componente podem ser editadas. Veja na figura a atualização da propriedade Time, que define o horário que indicará o relógio. Em seguida, os componentes são conectados. O E-Slate usa a metáfora de quebra-cabeça para representar as conexões compatíveis. Finalmente, o micromundo final a ser executado consistirá na combinação de componentes interligados por conexões.

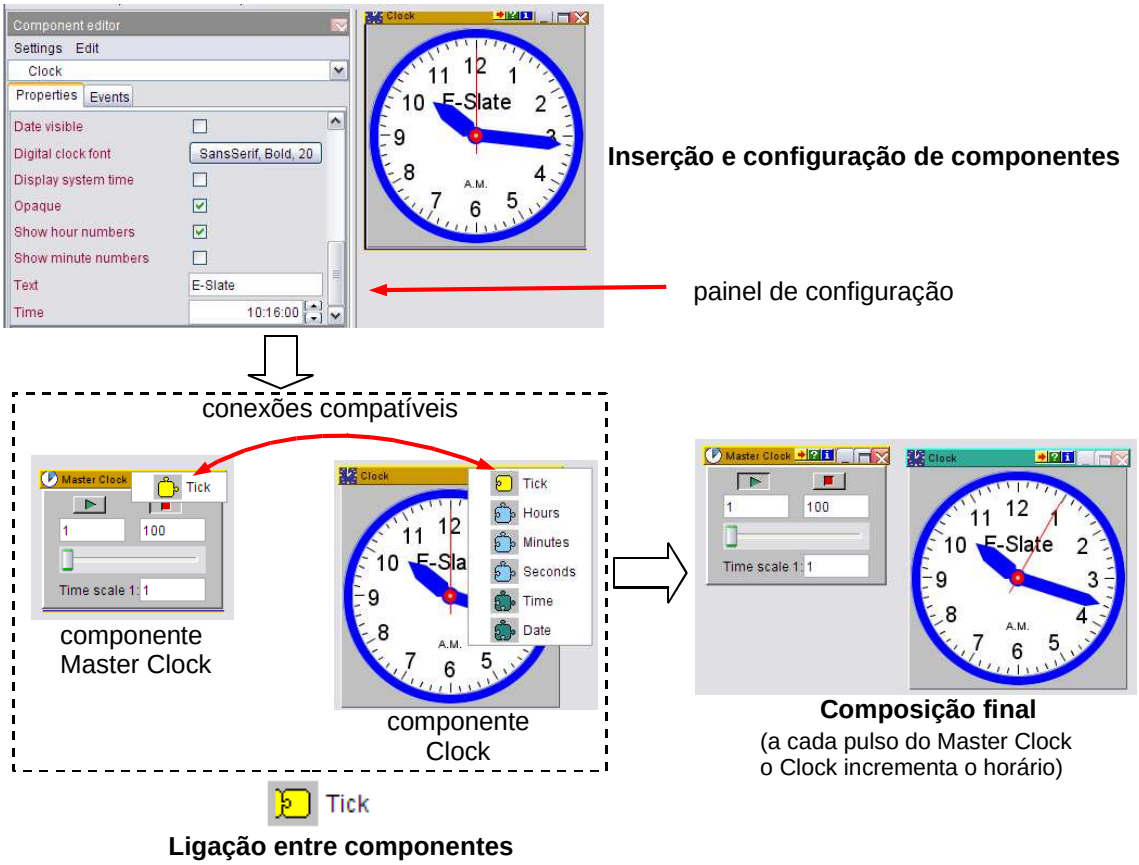


Figura 1. Processo de construção de um micromundo no E-Slate.

SimCalc e MathWorlds
http://www.simcalc.umassd.edu/
O MathWorlds, software integrante do projeto SimCalc, "[...] provê uma coleção de componentes de software, incluindo um conjunto de mundos de animação e uma variedade de gráficos. Os atores nos mundos (como um palhaço, ou um pato) movem-se de acordo com funções matemáticas. Os gráficos mostram essas funções matemáticas e permitem aos alunos editarem diretamente as funções".[Roschelle et al. 2000]. Atualmente o SimCalc e o MathWorlds são parte do mesmo sistema, ilustrado na Figura 2, que apresenta uma típica aplicação dentro do ambiente. Nele o autor pode configurar um cenário onde são dispostos atores (veja Figura 3), cujo movimento está descrito por funções matemáticas. O movimento na animação pode ser acompanhado também em gráficos cartesianos, tais como as de 'distância x tempo' e 'velocidade x tempo', apresentados nos gráficos abaixo do cenário. É possível modificar o movimento dos atores, através de uma intervenção direta em qualquer um dos dois gráficos. Modificações feitas com os atores também repercutem no gráfico. Como resultado, o aprendiz recebe subsídios que o permitem verificar, de modo bastante interativo, o papel do gráfico na descrição do movimento dos atores. O software fornece recursos para a criação de outros cenários, diferentes atores e movimentos a serem estudados. Cada uma das partes do ambiente (cenários, gráficos, tabelas etc.) é um componente de software, que o ambiente interliga para formar uma aplicação.

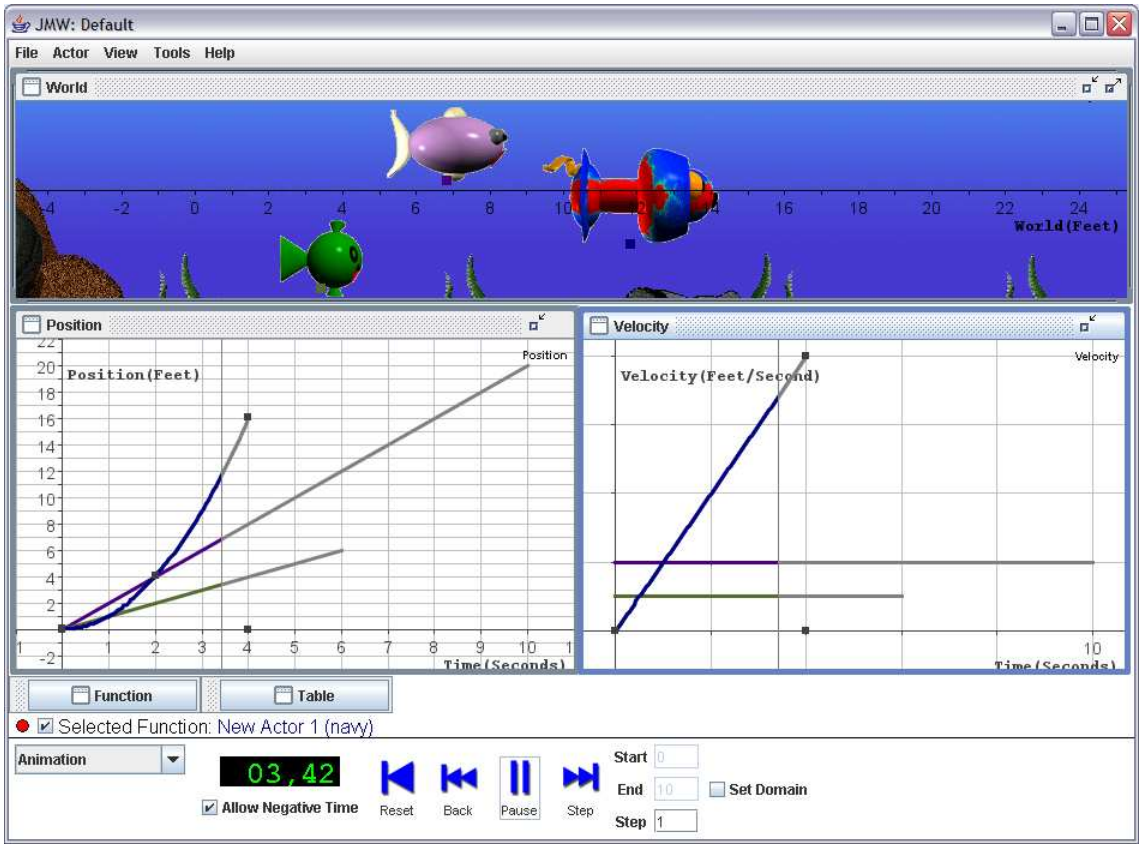


Figura 2. Animação feita no SimCalc de atores que se deslocam conforme funções matemáticas.

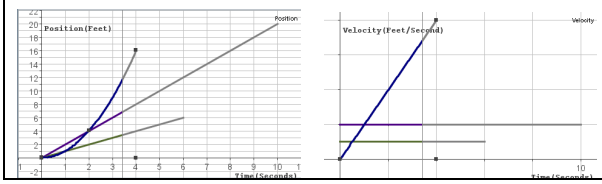
	Atores cujo movimento é descrito por funções matemáticas.
	Gráfico ilustrando a evolução da posição e velocidade dos atores no tempo.

Figura 3. Detalhamento de elementos envolvidos em uma animação do SimCalc.

Reuniram-se essas e outras iniciativas em um projeto ousado de nome ESCOT – *Educational Software Components of Tomorrow* – cujo objetivo é possibilitar interoperabilidade entre componentes educacionais de diversas origens, de modo que eles possam se integrar em aplicações que reúnam vários componentes [ROSCHELLE et al., 1999]. O ESCOT elegeu o modelo de componentes do Java para integração.

A meta é tornar os componentes educacionais blocos de construção básicos:

> *"No futuro, as idéias para atividades educacionais específicas vão converter-se em software tão fácil quanto a escrita de um documento: O processo de desenvolvimento será substituído por um processo de edição, criação e manipulação de 'aplicações editáveis'. Estas aplicações consistirão em objetos computacionais de alto nível, disponíveis como blocos tangíveis de construção." [Roschelle et al. 1999]*

As idéias defendidas pelos componentes educacionais serão novamente retomadas nos objetos de aprendizagem, ampliando-se para um universo que vai além do código executável, incluindo imagens, vídeos etc.

3. Objetos de Aprendizagem

Para tornar mais concreta a apresentação do conteúdo a seguir, optamos por utilizar um conjunto de exemplos que ilustrarão cada um dos conceitos apresentados. Na próxima subseção serão apresentados exemplos do que consideramos OA. Nas subseções e seções subseqüentes faremos referência e analisaremos estes exemplos.

3.1. Exemplos de Objetos de Aprendizagem

Objeto de Aprendizagem 1 – Animação da Fagocitose

Objetos de aprendizagem podem ser de vários tamanhos e complexidade. Iniciaremos trabalhando com um exemplo bastante simples. Trata-se de uma seqüência de imagens que apresenta etapas de uma célula englobando outro corpo, através de um movimento conhecido como fagocitose, conforme ilustrado na Figura 4. Tal seqüência está armazenada em um único arquivo de imagem.

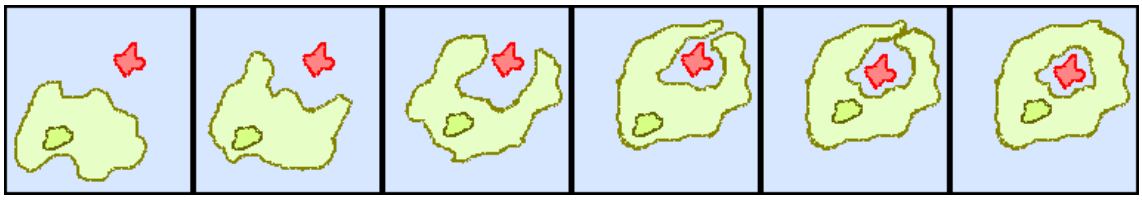


Figura 4. Exemplo de OA – seqüência de imagens ilustrando uma fagocitose.

Objeto de Aprendizagem 2 – Lados e Ângulos de um Polígono

Este segundo OA se apresenta com a faceta dos componentes de software. Ele segue a linha dos Componentes Educacionais apresentados na seção de precursores. Trata-se de um componente de software escrito em Java, cujo objetivo é fornecer um objeto interativo para demonstrar como os lados e ângulos se comportam em polígonos.

Mediante propriedades editáveis externamente, o componente pode assumir diversas configurações. A Figura 5 mostra o componente em três configurações diferentes. Em azul estão indicadas as medidas dos lados e em verde, a medida dos ângulos internos.

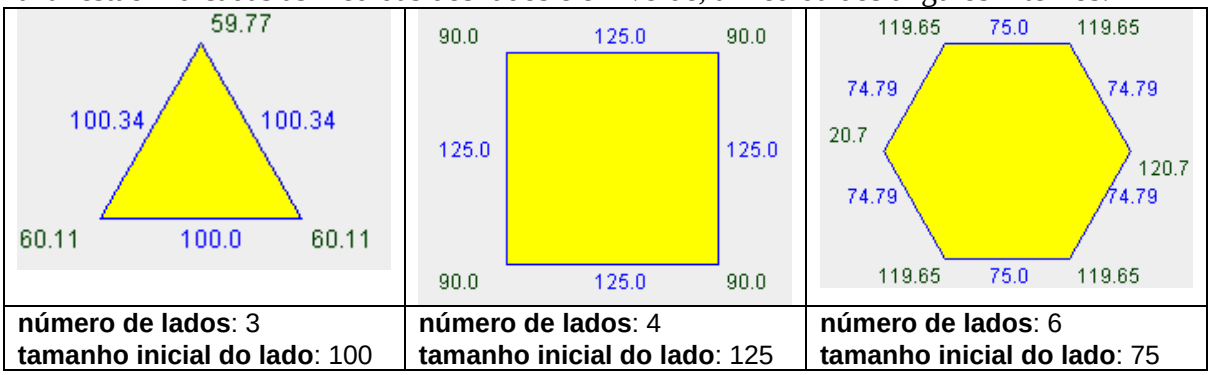


Figura 5. Exemplo de OA que assume três configurações para estudo de polígonos.

Como está ilustrado na Figura 6, o componente é interativo, ou seja, o usuário pode livremente clicar e arrastar os vértices do triângulo, a fim de observar o comportamento dos lados e dos ângulos. Fazendo isso, o usuário pode constatar, por exemplo, que a soma interna dos ângulos de um triângulo é sempre 180°.

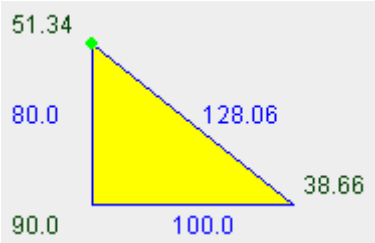


Figura 6. Usuário movendo os lados do polígono para criar um triângulo retângulo.

Objeto de Aprendizagem 3 – Corpo Humano

Enquanto os exemplos anteriores apresentaram objetos mais simples, que usualmente farão parte de produtos maiores, este terceiro exemplo vai além, partindo para um objeto mais complexo que contém uma aplicação completa. Este exemplo foi originalmente apresentado em Dourado et al. (2007).

A produção do objeto abaixo pode ser feita por um professor de ciências ou biologia para os seus alunos, ou com os seus alunos. Trata-se de um material educativo sobre a anatomia do corpo humano, concebido para ser executado em um navegador Web. A Figura 7 apresenta a página inicial deste OA no momento em que ele é apresentado em um navegador Web; consiste em um conjunto de páginas HTML interligadas por links. Como está indicado na figura, a página principal utiliza um recurso do HTML que associa links a regiões geométricas da página.

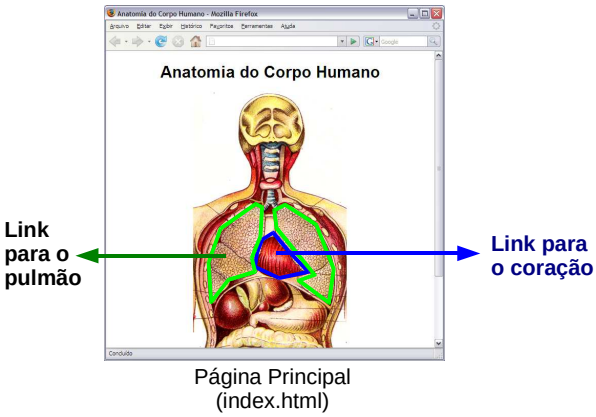


Figura 7. Exemplo de OA para exploração interativa da anatomia humana.

O aluno pode clicar em partes do corpo humano e, através de links mapeados na imagem, será conduzido a outras páginas que contêm detalhes sobre o respectivo órgão. Para fins de simplificação, apenas uma parte do objeto, detalhando o coração e o pulmão, foi incluída no exemplo apresentado.

A Figura 8 apresenta um diagrama de dependências entre os arquivos que compõem o objeto. Pode-se observar a página principal no topo que: utiliza uma imagem da anatomia humana, define dois links para outras duas páginas com detalhes sobre o

coração e o pulmão, e utiliza uma folha de estilo. A mesma interpretação pode ser usada para os demais arquivos.

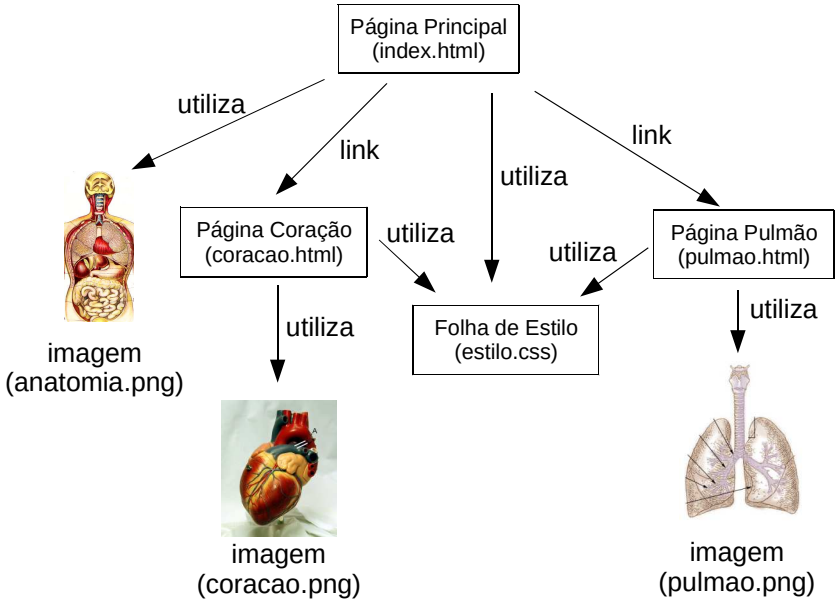


Figura 8. Arquivos que fazem parte do OA: anatomia e suas relações.

3.2. Características de um Objeto de Aprendizagem

A perspectiva do que representa software em geral - e em especial software educacional -, sofreu grandes modificações nos últimos anos. Gradativamente nos deslocamos de uma antiga noção de software fortemente amarrada àquela de código executável, para uma riqueza de possibilidades que inclui páginas Web, animações, produções multimídia. O código executável - que antigamente era produzido por alguns iniciados em computação - amplia suas possibilidades para múltiplas expressões de autores de todos os domínios. Hoje nos encontramos diante de um imenso repositório (a Web), compartilhado por alguns milhões de autores espalhados por todo o mundo. E daí surge a questão: Por que não utilizar toda essa riqueza de materiais já construídos? Por que sempre iniciar um novo material partindo do zero? Diante de um tempo cada vez mais escasso, da limitação de recursos e da encantadora diversidade de idéias e culturas expressas na forma de artefatos digitais, a prática do reuso se torna urgente.

Como veremos na próxima subseção, tal necessidade se tornou premente não apenas na comunidade da Educação, dando origem a iniciativas com propostas e características equivalentes em diferentes domínios.

Uma das dificuldades em encontrar uma definição precisa para OA está relacionada com a origem do termo. É difícil dizer com precisão qual a sua origem. As definições encontradas não são equivalentes e os primeiros autores a usar o termo não se referenciam, dando a impressão de que estavam adotando a terminologia para contextos muito específicos, sem a intenção de generalizar aquele. Alguns autores [Wiley 2000, von Brevern 2004] relacionam e confrontam diferentes abordagens e definições para OA, mas não nos parece possível encontrar qualquer indício de unicidade. Por este motivo, ao invés de lançarmos (mais) uma definição para o termo, traçaremos um conjunto de características que nos parecem essenciais para entender um OA.

Característica 1 – Unidade de reuso

Nosso ponto de vista é que a noção de OA está intimamente ligada à questão do reuso. Na definição mais popular para o termo (não a primeira), já encontramos claramente a questão do reuso:

> *"Objeto de aprendizagem é aqui definido como qualquer entidade, digital ou não digital, que pode ser usada, reusada ou referenciada durante o aprendizado apoiado sobre a tecnologia. Exemplos de aprendizado apoiado em tecnologia incluem sistemas de treinamento baseados no computador, ambientes de aprendizado interativo, sistemas inteligentes de instrução auxiliada por computador, sistemas de aprendizado à distância, sistemas de aprendizado baseados na Web e ambientes de aprendizado colaborativo"[IEEE L.T.S.C. 2000].*

É preciso cuidado ao tratar da questão do reuso, porque a palavra pode ter diferenças sutis de interpretação, mas extremamente relevantes no nosso contexto. O dicionário Houaiss,2006] define "reutilizar" (o mesmo que reusar [Houaiss 2006]) de duas maneiras: (i) "utilizar de novo" ou (ii) "imprimir novo uso a". O dicionário Aurélio [Ferreira 2004] também tem duas definições para o termo: (i) "tornar a utilizar" ou (ii) "dar novo uso a".

Note-se que a interpretação (i) de ambos os dicionários – reusar no sentido de usar de novo – é comum no português, mas é incomum no domínio da Informática. Por exemplo, não é usual dizer que "você está reusando o seu sistema operacional toda vez que liga o computador", ainda que você esteja usando-o novamente. Em geral, a palavra reuso em Informática é aplicada com a interpretação (ii) de ambos os dicionários, ou seja, dar um novo uso a algo. Por este motivo, a partir de agora o termo reuso será aplicado sempre com o significado (ii).

O reuso pode ser oportunístico ou sistematizado [Rockley 2000]. Chama-se de oportunístico o reuso ocasional, não-planejado, que geralmente usa técnicas de "copiar e colar". O reuso sistematizado, por outro lado, exige planejamento e preparo. Diversos aspectos têm que ser considerados para que ele aconteça de forma efetiva. Do ponto de vista técnico o reuso sistematizado de conteúdo exige: a definição de estratégias de como o conteúdo será reusado, a aplicação de padrões na sua produção, a documentação e indexação dos artefatos etc. Além disso, devem ser considerados aspectos de como o conteúdo será adaptado para diversos contextos, já que o reuso implica em dar um "novo" uso. Do ponto de vista pedagógico, a adaptação é um fator essencial. Como observa Roschelle et al.: "Aplicações educacionais têm que ser muito flexíveis porque o currículo e estilos de ensino variam tremendamente entre instituições, locais, e até mesmo entre instrutores da mesma instituição" [Roschelle et al. 1999]. Esta observação é reforçada e expandida por diSessa:

> *"Educação é fundamentalmente uma prática local e culturalmente específica. Há uma grande diversidade nas práticas dos professores, uma diversidade que a pesquisa mostrou nos remeter a uma diversidade no conhecimento dos professores sobre o assunto, seu repertório de estratégias para ensinar conceitos particulares (freqüentemente referenciado como conhecimento de conteúdo pedagógico), suas crenças sobre sua função como professores, suas crenças sobre aprendizes e aprendizado, e suas crenças sobre a natureza do conhecimento e o que eles acreditam que deve ser ensinado (FENNOMA & FRANKE, 1986; SHULMAN, 1986; THOMPSON, 1992). A menos que a tecnologia educacional seja adaptativa, os professores terão dificuldades em modificar seu estilo de trabalho, fruto de longa experiência, para ajustá-lo ao computador, ao invés do contrário." [di Sessa 2001]*

Concluímos que a capacidade de se adaptar é essencial para o reuso. Na esfera de desenvolvimento de software aprendemos que se torna tanto mais fácil adaptar um objeto, quanto menos precisarmos alterar seu conteúdo no processo de adaptação. Isso é possível pelo uso de parâmetros que configuram o objeto, tal como pode ser observado no Exemplo 2 de OA, no qual o número de lados do polígono e o tamanho de seus lados podem ser configurados. Esta é uma prática comum no contexto dos componentes de software, que geralmente podem ser configurados externamente pela atribuição de valores a propriedades.

A capacidade de configuração geralmente cresce de importância à medida que o objeto se torna mais complexo. Por exemplo, o Exemplo 1 de OA tem poucas opções de configuração, uma vez que se trata de um conjunto de imagens, no entanto, por ser um objeto bastante simples, ele geralmente será usado como peça de aplicações maiores e poderá ser usado tal como está. Quando o objeto é mais complexo, tal como o Exemplo 3 de OA, ele deixa de ser um simples bloco de construção e assume uma característica de aplicação com uma estrutura de aplicação e uma metodologia específicas. Neste caso, conforme já foi observado por Roschelle et al. (1999), e diSessa (2001), se a aplicação não for adaptável o bastante para diferentes contextos, ela provavelmente não será reusada.

Característica 2 – Digital

Ainda que a definição mais popular de OA inclua "qualquer entidade, digital ou não digital" [IEEE L.T.S.C., 2000], parece-nos sem sentido considerarmos as entidades não-digitais. Quando lidamos com computadores, tratamos sempre de entidade digitais e, por este motivo, praticamente todas as aplicações de OA encontradas lidam com entidades digitais [Sosteric e Hesemeier 2002].

Compreendemos esta inclusão das entidades não-digitais no contexto em que esta definição surgiu. Como veremos na próxima seção, ela foi cunhada pelo *Learning Technology Standards Committee* (LTSC) do IEEE, no contexto da definição de um padrão de metadados para OAs. É natural imaginar que metadados possam descrever algo não-digital, como um livro na prateleira de uma biblioteca. Mas mesmo neste caso, é preciso haver um mecanismo de identificação digital deste livro, para que possa ser feita a associação dos metadados. Tudo dentro do computador é tratado digitalmente, e não faz diferença para ele se o objeto final é digital ou não. Por este motivo, uma simplificação bastante conveniente é tratar todos os OAs como objetos digitais.

Característica 3 – Mais do que código executável

Uma pergunta recorrente de quem inicia na área de OA é se eles obrigatoriamente são programas de computador, ou seja, unidades feitas de código executável em alguma linguagem de programação. Nenhuma das definições originais – das mais antigas às mais populares – restringe os OAs a programas de computador [L'Allier 1997, IEEE L.T.S.C. 2000, Wiley 2000].

Com as múltiplas tecnologias disponíveis atualmente, é difícil explicar o que distingue um *software* executável de um *software* não-executável. Como classificar, por exemplo: composições hipermídia (e.g., aplicação em Flash), planilhas eletrônicas etc. Esta questão é abordada em detalhes em [Santanchè et al. 2007], no contexto do ambiente Casa Mágica.

Casa Mágica
`http://purl.org/NET/casamagica`
O sistema Casa Mágica [Santanchè and Teixeira 2001] é um ambiente para a construção de aplicações educacionais que combina os recursos de uma ferramenta de autoria com recursos que permitem a construção e exploração de modelos de estudo. Em suas primeiras versões, o sistema Casa Mágica utilizou componentes de software educacionais [Santanchè and Teixeira 2001] como base para a construção de aplicações, mas nos últimos ele ampliou suas possibilidades com a inserção dos Componentes de Conteúdo Digitais (Digital Content Components [Santanchè and Medeiros 2007, Santanchè et al. 2007]) que pode ser considerada uma generalização dos Objetos de Aprendizagem. Este sistema pode ser adotado nos mais diversos níveis educacionais e para as mais diversas áreas. Relatos de aplicações podem ser encontrados em [Santanchè 2002, Santanchè and Teixeira 2002].

A Figura 9 ilustra uma típica aplicação desenvolvida no sistema Casa Mágica [Santanchè et al. 2007], cujo objetivo é ilustrar a dinâmica do Movimento Retilíneo Uniforme (MRU) em uma atividade de física. Nela estão seis objetos (Componentes de Conteúdo Digital – v. próxima seção) interligados: um interruptor, um relógio, um contador, uma planilha, um cenário e uma esfera.

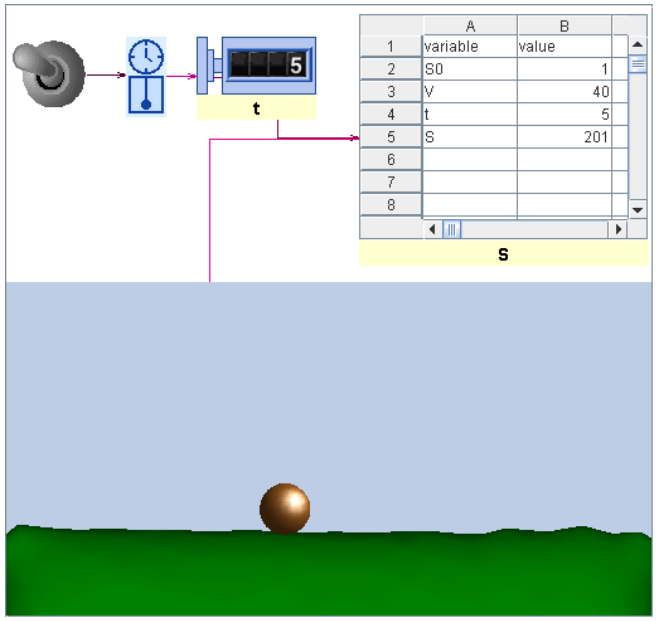


Figura 9. Simulação produzida no sistema Casa Mágica usando Componentes de Conteúdo Digital [Santanchè et al. 2007].

Resumidamente, a aplicação funciona da seguinte maneira: quando o usuário clica no interruptor ele manda uma mensagem de "inicia funcionamento" para o relógio, que por sua vez passa a mandar a cada segundo uma mensagem de "próximo número" para o contador (rótulo t). Este contador é uma representação discreta da contagem do tempo: cada vez que recebe uma mensagem do relógio, ele incrementa seu valor e envia uma mensagem à planilha, para que ela atualize o valor correspondente à célula B4, que corresponde ao valor de "t". A planilha foi preparada para representar e calcular um MRU por meio da fórmula $S = S_0 + V.t$. Cada linha possui uma constante ou variável da fórmula com o respectivo valor. Na linha de S, existe uma fórmula para calcular V a partir dos outros valores cada vez que "t" é atualizado. A planilha envia mensagens para a esfera, de modo que ela atualize sua posição conforme a variação de S.

Esse exemplo ilustra a dificuldade em se diferenciar o que é código executável do que não é. Essa aplicação combina objetos tipicamente considerados como código executável, tais como o relógio e o contador, com outros, como a planilha eletrônica. Apesar do software que processa a planilha ser código executável, a planilha em si geralmente não é classificada desta forma, ainda que se possam criar planilhas que se comportam como se comportariam programas de computador.

Característica 4 – Aderência a padrões abertos

Para que seja possível o reuso de qualquer artefato digital é necessário algum tipo de padrão aberto, já que o reuso prevê um novo contexto que não aquele onde o artefato foi concebido. Note-se que não se tratam necessariamente de padrões abertos específicos para OA. Por exemplo, hoje é possível compartilhar-se e reusar qualquer imagem através da Web, graças a padrões abertos de representações de imagens, tais como: GIF, JPEG, PNG etc. Antes deles isto era difícil ou inviável. Da mesma maneira, quando falamos de reuso como uma característica inerente a um OA, precisamos tratar de padrões. A questão é: Quais padrões? Se não existe uma definição precisa do que é um OA, também não é possível exigir-se que um OA adote necessariamente um padrão x ou y. Seria o mesmo dizer que um arquivo para ser imagem tem que ser sempre GIF; isso excluiria todos os demais padrões de compartilhamento de imagens.

Se por um lado não podemos associar a definição de OA a um padrão específico, por outro isto não nos isenta de escolher algum padrão entre as opções disponíveis. Os padrões para OA atuam diretamente sobre outros padrões, como aqueles usados para mídias, enriquecendo-os do ponto de vista educacional com: metadados, organização de tópicos etc. Os padrões serão tratados com mais detalhes na Seção 4.

Nenhuma definição popular de OA inclui explicitamente a questão de aderência a padrões abertos. Por outro lado, todas essas definições são encontradas em textos ou relacionados com a definição de padrões abertos, ou que discutem padrões abertos para OAs. Isto demonstra o forte vínculo entre a noção de OA e os padrões.

Característica 5 – Voltados à aprendizagem

Esta é uma característica derivada diretamente do nome. Por desnecessário que pareça destacá-la, ela (aparece para) ressalta que um OA não se resume a um conjunto de padrões técnicos, mas deve considerar principalmente os aspectos pedagógicos. De fato, uma das primeiras definições de OA (senão a primeira) o caracteriza sob o ponto de vista pedagógico "definido com a menor experiência instrucional independente que contém um objetivo, uma atividade de aprendizagem e uma avaliação"[1] [L'Allier 1997].

Como veremos na próxima seção, a preocupação com os aspectos pedagógicos são um diferencial dos padrões abertos para OA em relação aos outros padrões.

4. Padrões Abertos para Objetos de Aprendizagem

O principal segmento de pesquisas de padrões abertos relacionados com OAs tem suas origens a partir dos trabalhos do IEEE *Learning Technology Standardization Committee* (LTSC) (`http://ieeeltsc.org`). Este comitê é uma iniciativa para o

[1] Tradução feita pelo autor do original: "[...] *we call it the Learning Object structural component, which is defined as the smallest independent instructional experience that contains an objective, a learning activity and an assessment*." [L'Allier 1997]

desenvolvimento de tecnologias educacionais baseadas em padrões abertos de interoperabilidade. Especificamente o grupo de trabalho 12 (WG12), incumbido de definir do padrão de metadados, foi o responsável pela definição mais popular do termo *Learning Object* ou Objeto de Aprendizagem, em sua especificação para o LOM – *Learning Object Metadata* [IEEE L.T.S.C. 2000, IEEE L.T.S.C. 2002]. Associados a este padrão inicial, surgiram diversos padrões relacionados para:

- Metadados
- Empacotamento
- Interfaces e comunicação
- Seqüenciamento
- Integração dos demais padrões

Nas subseções a seguir analisaremos cada um destes padrões.

4.1. Metadados

Com uma vasta e crescente quantidade de OAs disponíveis na Web, a questão passa a ser como classificar e documentar tais objetos, de maneira que se possam projetar ferramentas aptas a encontrá-los e utilizá-los.

Learning Object Metadata

Com este objetivo, o LTSC desenvolveu o padrão LOM [IEEE L.T.S.C. 2002]. O padrão IEEE LOM deriva de trabalhos anteriores desenvolvidos na fundação européia ARIADNE (`http://www.ariadne-eu.org`), e pelo IMS Global Learning Consortium (`http://www.imsglobal.org`) [Nilsson et al. 2006].

No padrão LOM, os OAs são descritos por meio de um conjunto de propriedades, às quais são dados valores que caracterizam o objeto. O padrão define um esquema que relaciona hierarquicamente as propriedades aceitas, o domínio dos valores que elas podem receber, bem como sua cardinalidade. Este esquema, denominado *BaseScheme* (esquema básico), reúne as propriedades consideradas fundamentais para a descrição dos objetos. Ele pode ser estendido para outros esquemas derivados, de acordo com necessidades específicas.

A estrutura geral do *BaseScheme* é dividida segundo grupos de propriedades:

General	Dados de identificação, chaves de acesso, delimitação do domínio, descrição do conteúdo e da estrutura.
Life Cycle	Dados para controle e documentação do ciclo de vida do documento.
Meta-MetaData	Referências à origem e estrutura dos metadados.
Technical	Dados técnicos, tais como: formato, tamanho, requisitos de sistema operacional, duração, etc.
Educational	Elementos de descrição pedagógica do recurso, tais como: abordagem, nível de interatividade, pré-requisitos, objetivo educacional, grau de dificuldade, tempo esperado de trabalho, etc.
Rights	Dados referentes às condições de uso do produto e, eventualmente, valores a serem pagos pelo uso do recurso.

Relation	Características que relacionam o objeto de aprendizagem com outros.
Annotation	Comentários referentes ao uso educacional do produto.
Classification	Referência que determina onde o recurso será colocado dentro de um sistema de classificação específico.

Recentemente, o IEEE LTSC mapeou o esquema do LOM para XML [IEEE L.T.S.C. 2005], e o trabalho foi baseado em um padrão anterior encabeçado pelo consórcio IMS [Thropp and McKell 2001]. Para ilustrar como o LOM é criado e usado, apresentaremos um exemplo prático de elaboração de alguns metadados relacionados ao Exemplo 1 de OA. Para fazer isso será utilizada a ferramenta Reload Editor.

Ferramentas do projeto RELOAD

`http://www.reload.ac.uk`

O projeto RELOAD (Reusable eLearning Object Authoring & Delivery) tem como meta desenvolver ferramentas que oferecem suporte às tecnologias educacionais recentes que definem padrões interoperáveis; estão nesta lista diversos padrões que serão analisados neste texto: IEEE LOM, IMS CP, SCORM, etc.

Atualmente, o projeto possui cinco ferramentas:

- Metadata and Content Package Editor (Reload Editor)
- SCORM Player
- Learning Design Editor
- Learning Design Player

A Figura 10 apresenta uma tela de edição dos metadados educacionais segundo o padrão LOM (note-se que é necessário escolher a opção LTSN Profile, conforme está indicado na ilustração). Como é usual nas ferramentas para este padrão, é apresentado um formulário ao usuário, no qual cada campo corresponde a uma propriedade do LOM.

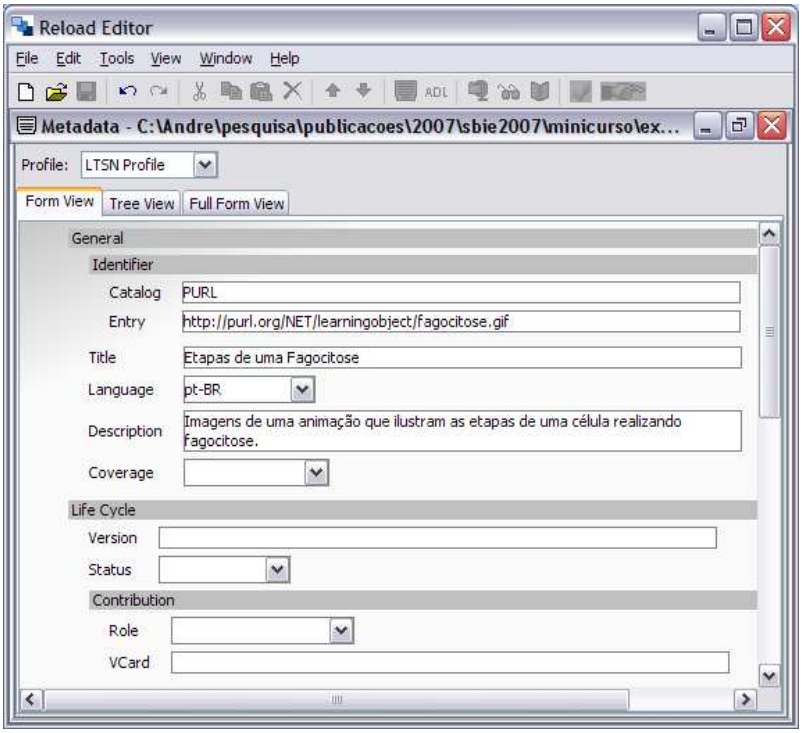


Figura 10. Tela de edição de metadados LOM do Reload Editor.

Neste exemplo apenas os seguintes campos foram preenchidos:

Campo	Descrição
Identifier **Catalog**: Padrão catalográfico escolhido. **Entry**: Identificação única conforme o padrão catalográfico.	Identificação única do objeto. Recomenda-se que não existam dois objetos dentro do contexto desejado com o mesmo identificador.
Title	Título do objeto que será lido pelo usuário.
Language	Linguagem em que os metadados estão descritos (pt-BR indica português do Brasil).
Description	Descrição do objeto que será lida pelo usuário.

Os seguintes valores foram inseridos nos campos

Campo	Descrição
Identifier	**Catalog**: PURL **Entry**: `http://purl.org/NET/learningobject/fagocitose.gif`
Title	Etapas de uma Fagocitose
Language	pt-BR
Description	Imagens de uma animação que ilustram as etapas de uma célula realizando fagocitose.

O ideal é que o identificador criado seja único no mundo. Mas como conseguir isto?

Uma alternativa amplamente adotada é o uso de um URI – *Uniform Resource Identifier*, que estabelece um meio de identificar qualquer recurso de forma única dentro da Web. O tipo mais popular de URI é um subconjunto denominado URL – *Uniform Resource Locator* –, que identifica seus recursos por meio de sua localização física na Internet; ele é usado, por exemplo, para: páginas na Web, endereços de e-mail, diretórios ou arquivos para FTP, etc. Por outro lado, o URI envolve um conjunto mais abrangente, em que se podem definir identificadores igualmente únicos na rede, mas que não estão associados diretamente à localização física do recurso. Por intermédio de serviços de nomes, o identificador é relacionado com sua localização física na rede. Este mecanismo preserva o identificador, mesmo que o recurso mude sua localização na rede ou que o recurso não esteja mais disponível na mesma.

Este é o caso do identificador do exemplo cujo prefixo (`http://purl.org/NET/learningobject`) foi criado em um servidor de nomes chamado PURL (`http://purl.org`). O uso do PURL está entre as recomendações do CanCore [Friesen et al. 2004]. O CanCore (`http://www.cancore.ca`) é uma iniciativa canadense cujo objetivo é potencializar a busca e localização de conteúdo educacional. Ele define um conjunto de recomendações (CanCore Guidelines) [Friesen et al. 2004] baseadas no IEEE LOM. Para maiores detalhes veja [Dourado et al. 2007].

O arquivo produzido em XML para este exemplo é:

```
<lom ...>
  <general>
    <title>
      <langstring xml:lang="pt-BR">
          Etapas de uma Fagocitose
      </langstring>
    </title>
    <catalogentry>
      <catalog>PURL</catalog>
      <entry>
        <langstring xml:lang="pt-BR">
           http://purl.org/NET/learningobject/fagocitose.gif
        </langstring>
      </entry>
    </catalogentry>
    <language>pt-BR</language>
    <description>
      <langstring xml:lang="pt-BR">
          Imagens de uma animação que ilustram as etapas
          de uma célula realizando fagocitose.
      </langstring>
    </description>
  </general>
</lom>
```

Note-se que o mapeamento para XML usualmente ocorre da seguinte maneira: para um campo de nome “catalog” cujo valor é “PURL”, o mapeamento resulta em:

```
<catalog>PURL</catalog>
```

Quando o campo deve ser lido pelo usuário, usa-se o marcador `<langstring>` com um atributo que indica em que língua está escrito o texto (isso permite criar textos multilíngües).

Apesar das evidentes vantagens desse mapeamento, o XML foi concebido como um padrão para definir uma estrutura e um formato universais de representação de dados, ou seja, trata-se de um padrão de interoperabilidade sintática. O padrão RDF – *Resource Description Framework* – da Web Semântica atua de forma complementar ao XML, proporcionando interoperabilidade na forma como estes dados são interpretados pelas aplicações (interoperabilidade semântica). O RDF define um modelo e uma linguagem para a representação homogênea de informações associadas a recursos que podem ser identificados por meio da Web [Manola 2003]. Tal representação é propícia para ‘consumo por computadores’ e atua diretamente sobre problemas de interoperabilidade.

O grupo de trabalho 12 (WG12) do LTSC indica a necessidade de uma futura compatibilidade do LOM com os padrões da Web Semântica, como está registrado na seção “LOM Next” do grupo (*http://www.ieeeltsc.org/working-groups/wg12LOM/LOMnext/*). Adicionalmente, Nilsson [Nilsson et al. 2003] apresenta uma proposta de mapeamento do LOM para RDF.

4.2. Empacotamento

Como pode ser observado no Exemplo 3 de OA, e especialmente na Figura 8, um objeto pode ser constituído de um conjunto diversificado de artefatos digitais, relacionados entre si de diversas maneiras. Isso conduz à questão: como organizar tal conteúdo em uma só unidade para facilitar sua distribuição e reuso? Esta é a questão central tratada pelos padrões de empacotamento.

Mediante o processo de empacotamento, um OA (com seus metadados associados) é armazenado dentro de um pacote. Um pacote é uma estrutura responsável pela agregação de múltiplos artefatos digitais em um arquivo único, preservando sua organização.

IMS CP

O IMS CP (IMS Content Package) é um modelo para empacotamento de OAs definido pelo IMS *Global Learning Consortium* [Smythe and Jackl 2004].

A estrutura de um pacote de IMS CP é definida conforme a Figura 11. Na figura, a estrutura externa, indicada como Package, consiste em um arquivo compactado em formato ZIP. Dentro do ZIP, está um arquivo especial denominado manifesto (*manifest*) com quatro subdivisões. Este arquivo será comentado adiante. Além do manifesto, ficam dentro do ZIP todos os demais artefatos que compõem o OA.

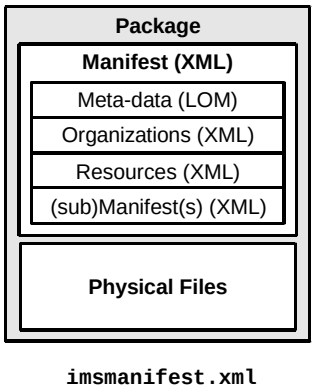


Figura 11. Estrutura do pacote IMS CP [Smythe and Jackl 2004].

Para se criar um pacote IMS CP pode ser usado o mesmo Reload Editor da subseção anterior. Na Figura 12 é apresentada a tela de edição do Reload Editor durante o processo de criação de um pacote IMS CP para o Exemplo 3 de OA.

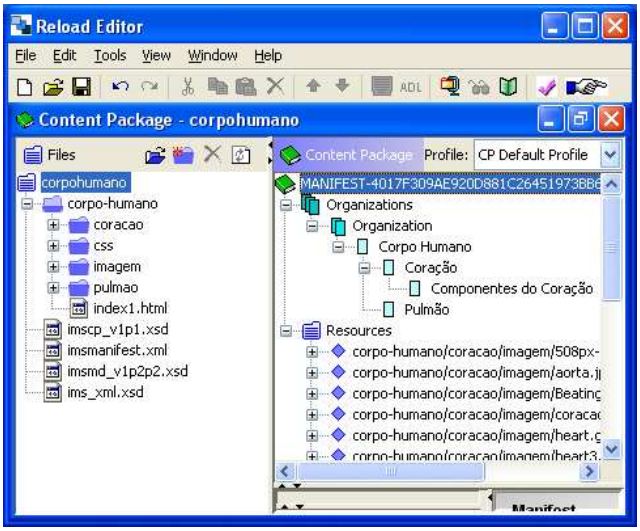


Figura 12. Reload Editor utilizado para criar um pacote IMS CP.

À esquerda da figura aparece a organização dos artefatos dentro do arquivo ZIP, à direita é detalhado o arquivo de manifesto. O destaque para o arquivo de manifesto

ressalta a sua importância como estrutura complementar para a organização dos artefatos.

O formato ZIP não apenas reúne todos os artefatos em um único arquivo, como é capaz de reproduzir a estrutura hierárquica dos diretórios que compõem o objeto. A Figura 13 apresenta o pacote IMS CP da Figura 12 aberto por um programa de manipulação de arquivos ZIP. Como é possível observar, o formato IMS CP utiliza o formato ZIP sem modificações. Todas as informações adicionais relativas aos artefatos se concentram no arquivo de manifesto, que aparece na raiz do arquivo compacto (`imsmanifest.xml`).

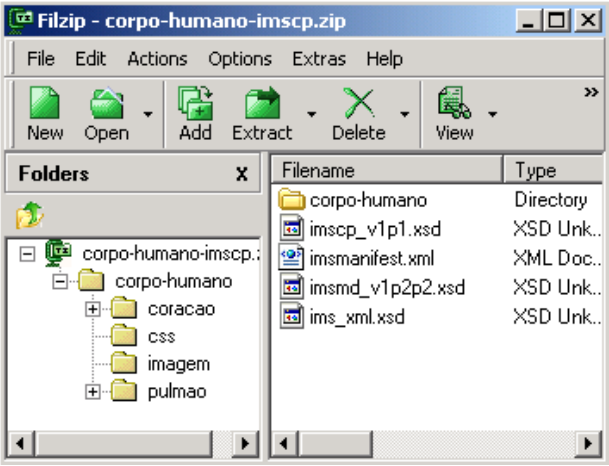


Figura 13. Pacote IMS CP aberto em um programa para formato ZIP.

No padrão IMS CP o manifesto é dividido nas seguintes seções (Figura 11):

- **Meta-Data** – metadados educacionais conforme o padrão LOM, que descrevem o objeto como um todo.
- **Organização** – define uma estrutura organizacional hierárquica que funciona como um índice de tópicos e subtópicos associados ao conteúdo educacional.
- **Recursos** – contêm referências para os artefatos que estão armazenados no ZIP e mapeiam as dependências entre eles.
- **Submanifestos** – esta seção é opcional, e contém manifestos subordinados quando há pacotes dentro de pacotes.

A seguir, apresenta-se outro exemplo prático de aplicação do IMS CP.

Considere um professor de biologia que está navegando pela Web, pesquisando conteúdo para a sua próxima aula sobre Fotossíntese. Durante a pesquisa encontra o site MIT OpenCourseWare[2], no qual estão disponíveis recursos gratuitos para estudantes e professores. Ao acessar o site, ele visualizará a página ilustrada na Figura 14. Na parte esquerda, ele encontra várias áreas, como: arquitetura, biologia, química etc. e escolhe sua área de interesse – Biologia. O link conduz para a página ilustrada na Figura 15.

[2] Massachussets Institute of Technology – OpenCourseWare - http://ocw.mit.edu/index.html

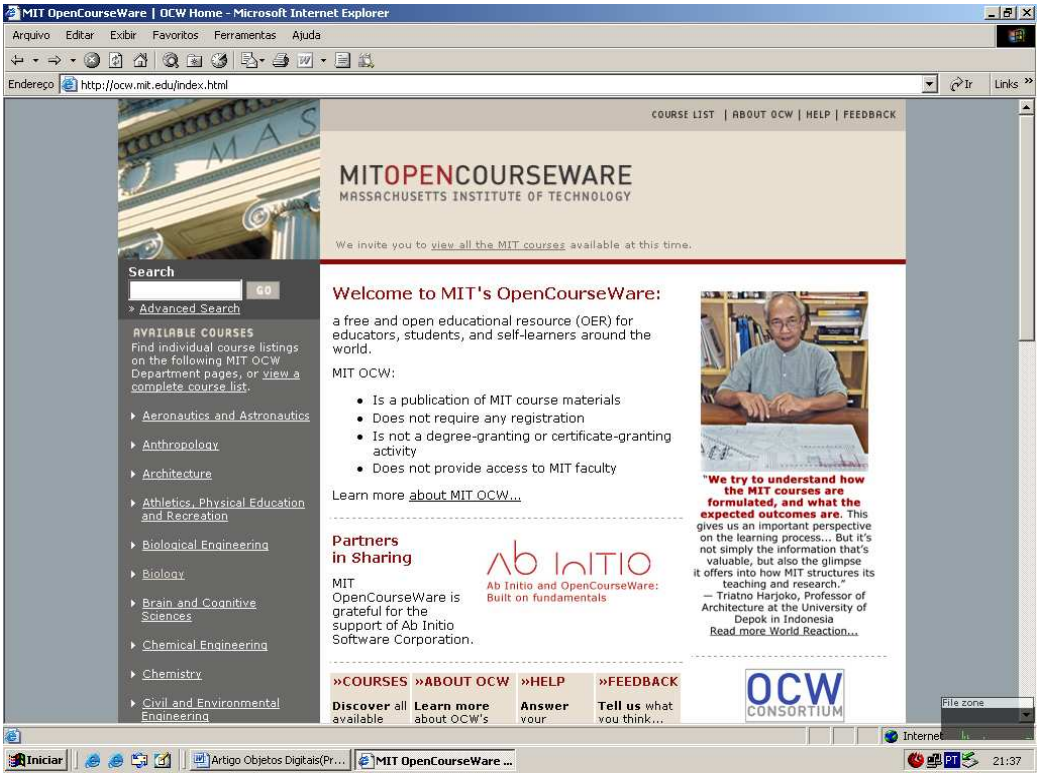


Figura 14. Página Inicial do MIT OpenCouseWare.

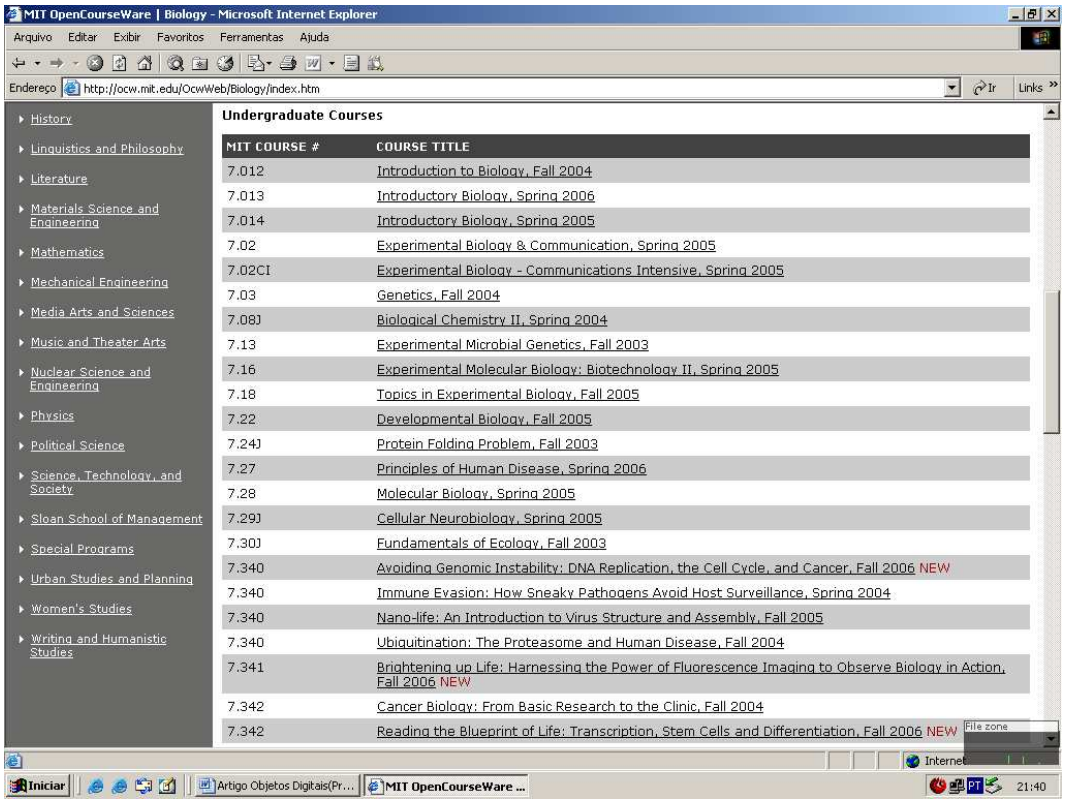


Figura 15. Página do MIT OCW referente ao tópico Biologia.

Em seguida, o professor seleciona um dos cursos disponíveis na lista apresentada. Considere-se que ele selecionou o curso de Fotossíntese. Será apresentada a página da Figura 16. Na figura à esquerda é mostrado o padrão de organização de um curso no MIT OpenCourseWare, que é padronizado para todos os cursos:

- **Course** Home – faz uma apresentação inicial, informando uma breve descrição do curso.
- **Syllabus** – sumário da programação do curso.
- **Calendar** – calendário do curso.
- **Readings** – o assunto Fotossíntese é apresentado em tópicos e, para cada um destes, são indicadas referências relevantes.
- **Lecture Notes** – notas para as aulas.
- **Assignments** – são disponibilizados alguns exercícios para serem executados.
- **Download this Course** – opção de fazer download do curso na forma de um pacote IMS CP.

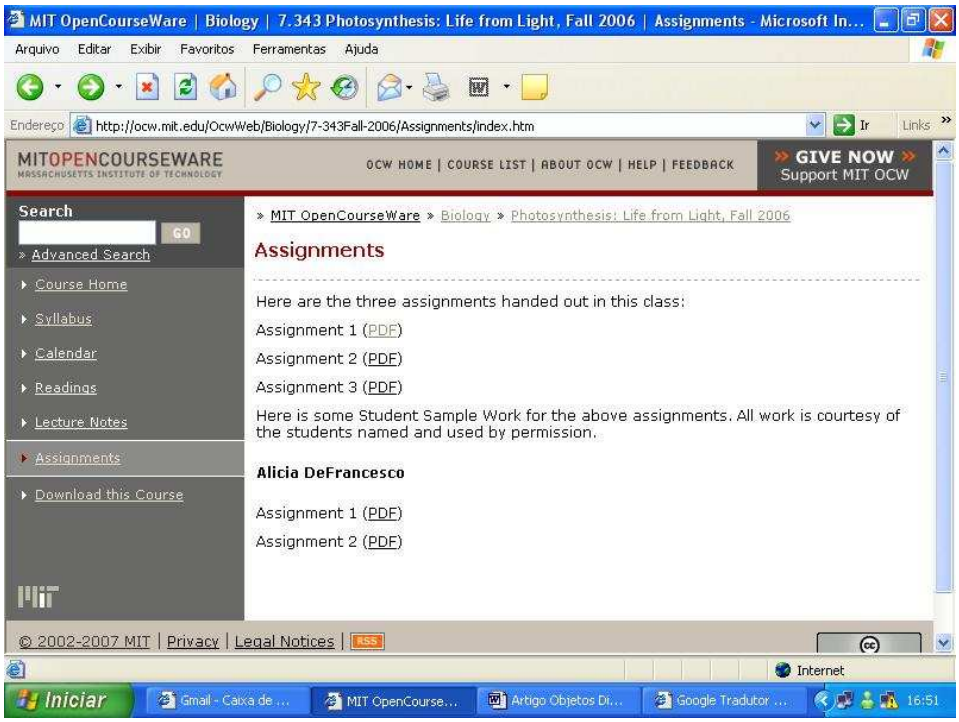


Figura 16. Página de um curso de Fotossíntese disponível no MIT OCW.

Pode-se constatar que o conteúdo de um curso é formado por um conjunto diversificado de artefatos inter-relacionados: além das páginas Web, com suas figuras, estilos e scripts. Há ainda material associado em outros formatos – veja ainda que a Figura 16, à direita, mostra a seção "Assignments", na qual o professor disponibilizou alguns arquivos PDF associados a uma determinada página Web.

Esse conteúdo pode ser caracterizado como um OA**.** Assim, para que o conteúdo possa ser compartilhado e reusado de forma consistente, é necessário que todos os itens que fazem parte da página estejam organizados em uma estrutura que mostre a dependência entre estes, bem como a sua organização. Por isso se usou o IMS CP. Conforme foi

feito no exemplo anterior, a Figura 17 apresenta o Reload Editor editando o pacote do MIT OCW.

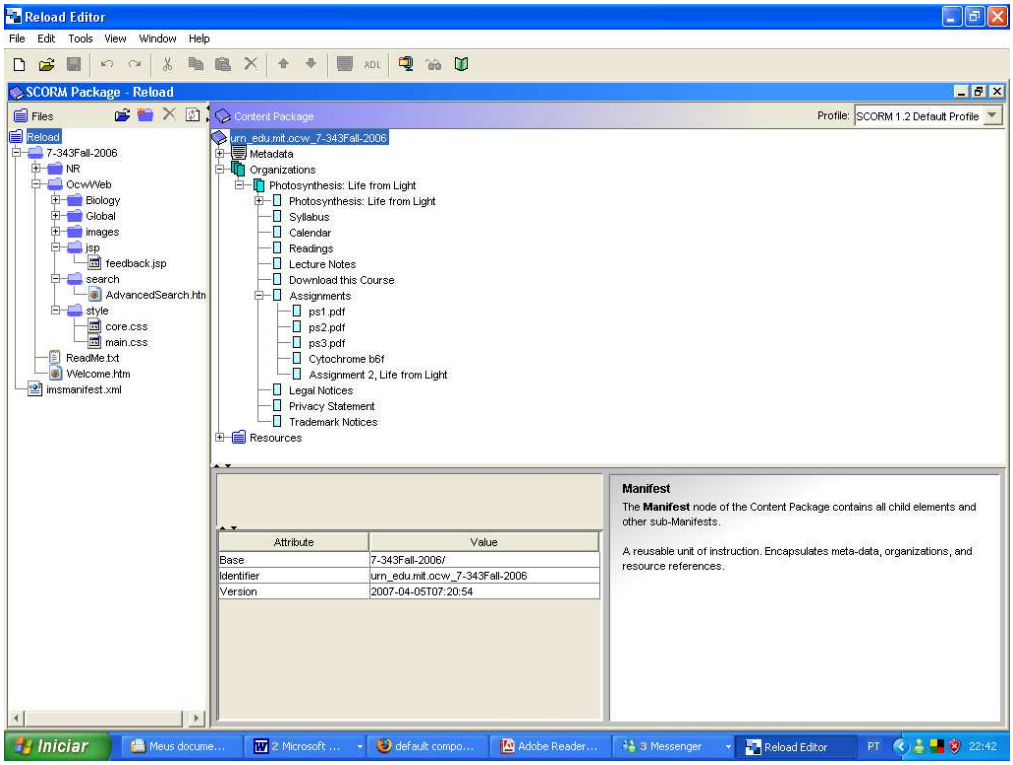


Figura 17. Pacote IMS CP do MIT OCW aberto no Reload Editor.

4.3. Comunicação com Sistemas de Gerenciamento de Cursos

Considerando-se que os OAs podem ser usados em atividades de aprendizagem gerenciadas por sistemas computacionais, são necessários padrões para que os objetos se comuniquem com o ambiente.

AICC CMI Guidelines

O *Aviation Industry CBT Committee* – AICC (`http://aicc.org`) definiu um padrão para a comunicação de módulos educacionais com ambientes de aprendizagem baseados na Web [McDonald et al. 2004]. Este padrão define API baseada em ECMAScript (uma generalização do JavaScript). Através desta API um OA pode solicitar serviços e informações do ambiente de aprendizagem e vice-versa.

4.4. Integração

Com tantos padrões disponíveis surge a necessidade de integração. Este é o propósito do SCORM.

SCORM

O SCORM – *Sharable Content Object Reference Model* [ADL 2006] é uma iniciativa da *Advanced Distributed Learning* – ADL (`http://www.adlnet.gov`) que unificou o IEEE LOM com o IMS CP e a especificação da AICC para interfaces e regras. O resultado é uma infra-estrutura com suporte para reusabilidade, interoperabilidade, localização e adaptação. O padrão integrado reúne:

- Metadados com uma extensão e adaptação do IEEE LOM.
- Empacotamento com uma extensão e adaptação do IMS CP.
- Comunicação com uma extensão e adaptação do AICC.

As adaptações são importantes, principalmente para combinar e compatibilizar os padrões entre si. A iniciativa ADL distribui um "Conformance Test Suite" para garantir que novos produtos sigam o modelo de referência.

4.5. Objetos de Aprendizagem nos Sistemas para Gerenciamento de Cursos

Vamos retomar o exemplo da Subseção 4.3 do MIT OCW para ilustrar o uso de OAs dentro de um Sistema para Gerenciamento de Cursos (SGC). Considere que, depois de fazer download do pacote, o professor do nosso exemplo deseja disponibilizá-la em um SGC, no nosso exemplo utilizaremos o Moodle.

O MOODLE – (**M**odular **O**bject **O**riented **D**istance **LE**arning) é um software de fonte aberta (Open Source Software), o que significa que se pode instalar, usar, modificar e mesmo distribuir o programa (nos termos da GNU General Public Licence). O Moodle pode ser usado, sem modificações, em Unix, Linux, Windows, Mac OS e outros sistemas de suportem PHP. Ele está disponível em 40 idiomas.

Como um SGC, o Moodle disponibiliza ferramentas de produção, comunicação e gerenciamento das interações dos usuários. Sua plataforma dispõe de várias ferramentas de comunicação, como o forum e o chat. Para produções coletivas possui a ferramenta WIKI, enquetes, glossários e base de dados que suportam vários usuários ao mesmo tempo. Na perspectiva de gerenciamento, é possível visualizarde forma simples as entradas e as participações dos usuários. O Moodle se apresenta como um interessante ambiente virtual de aprendizagem, que pode auxiliar o professor na produção de cursos, grupos de estudos, ambientes de simulações, no sentido de promover e potencializar espaços-tempo de aprendizagem.

Vejamos o uso do pacote IMS CP vinculado ao Moodle. Para isso o professor irá seguir os seguintes passos:

- Escolher a opção "Usar um pacote IMS CP":

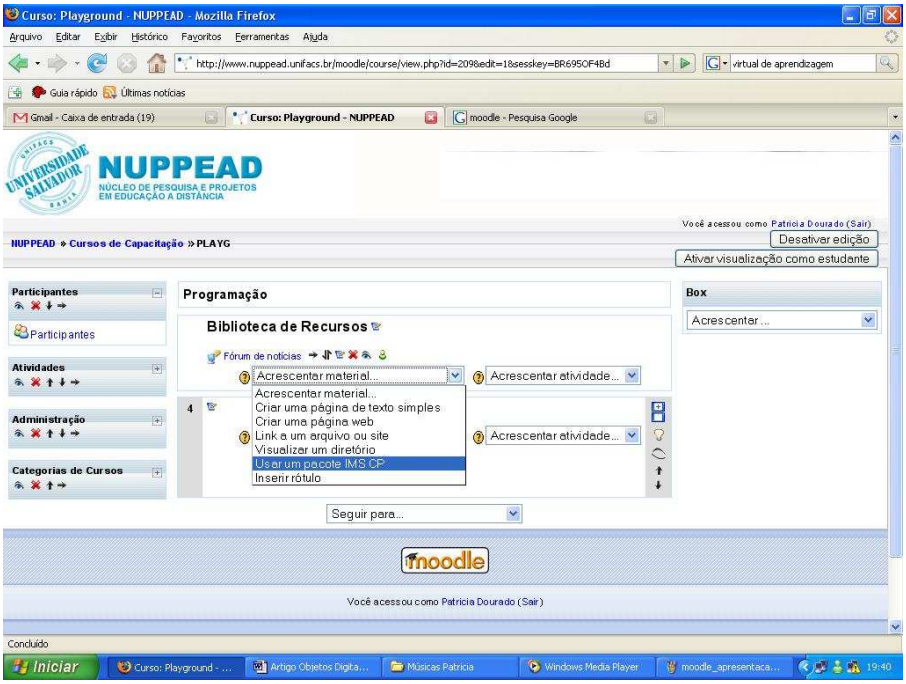


Figura 18. Início do processo de inserção do pacote IMS CP no Moodle.

- Importar o pacote e nomeá-lo:

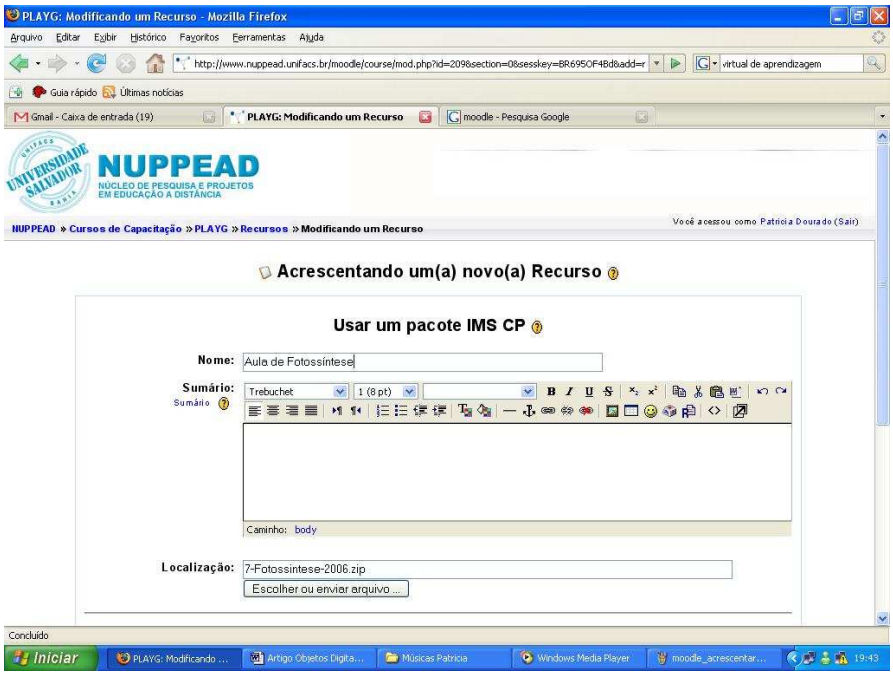


Figura 19. Tela de detalhamento dos dados do pacote no Moodle.

- Usar o pacote:

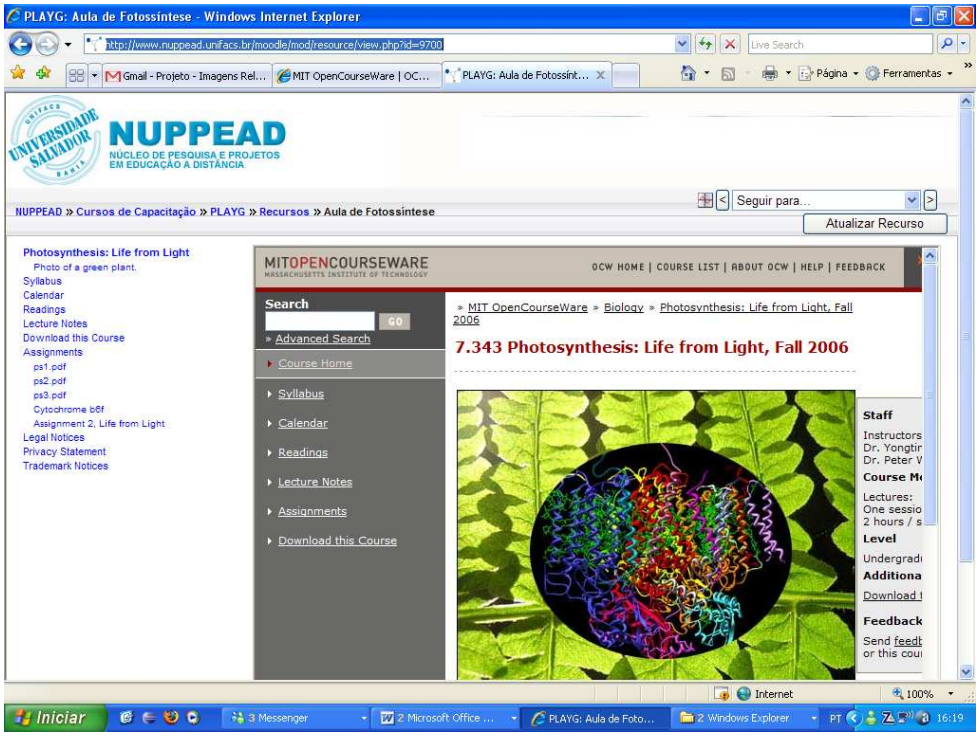


Figura 20. Pacote IMS CP do MIT OCW dentro do Moodle.

Retomando a Figura 17, na qual podemos ver em detalhes como se encontra a estrutura de organização interna no pacote, verificamos que a mesma organização se replica no Moodle na Figura 20 (menu apresentado à esquerda). Desse modo, a estrutura definida originalmente pelo MIT OCW se replica na estrutura de organização do pacote, e novamente no Moodle.

5. Objetos Digitais Complexos

É comum dentro do domínio da Informática aplicada à Educação considerar a noção de OA, original e única. O objetivo desta seção é demonstrar que, ao contrário, os OAs resultam de uma demanda igualmente percebida em outros domínios, que também têm desenvolvido pesquisas para a construção de tecnologias com características bastante similares às de um OA.

Partindo-se de uma perspectiva mais ampla, a questão mais importante a ser tratada é de como documentar (metadados), empacotar, distribuir, armazenar e catalogar artefatos de conteúdo digital de diversas naturezas, para posteriormente buscá-los e recuperá-los. Este é um dos temas principais tratados no domínio de bibliotecas digitais [Fox et al. 1995]. Tais artefatos, em geral, se caracterizam pela sua heterogeneidade e por serem constituídos de uma coleção complexa e inter-relacionada de subpartes – eles são objetos digitais complexos [Bekaert et al. 2005] [Nelson et al. 2001].

Ainda que sob diferentes denominações, os objetos digitais complexos têm sido tema de pesquisa em diversos domínios. Cada vez mais, torna-se necessária a criação de padrões relacionados a objetos digitais complexos para: empacotamento e distribuição, descrição através de metadados, apresentação do conteúdo, controle de autoria e direitos de uso etc. Os objetos de aprendizagem são um típico exemplo de objeto digital complexo. Além dos OAs, diversas são as pesquisas desenvolvidas em outros domínios:

Multimídia: Torna-se sempre mais necessária a distribuição de conteúdo multimídia, constituído pela combinação de artefatos das mais diversas naturezas. Para isso, o padrão MPEG-21 se propõe a atuar como uma cola [Burnett et al. 2003], por meio de uma especificação que permite a unificação de diferentes artefatos multimídia em um único pacote, preparado para a distribuição e consumo. Objetivando a apresentação/execução do conteúdo do pacote, o MPEG-21 está projetando uma infra-estrutura capaz de integrar diversas tecnologias multimídia, de modo que tudo seja feito de forma transparente para o usuário.

Desenvolvimento de Software: A OMG (Object Management Group) definiu o RAS – *Reusable Asset Specification* [OMG 2005], – com o objetivo de definir um padrão de empacotamento e descrição de artefatos digitais usados no processo de desenvolvimento de *software*.

Bibliotecas Digitais e Preservação a Longo Prazo: Uma das questões a serem consideradas ao se armazenarem objetos digitais complexos em bibliotecas é o impacto causado pelas mudanças das tecnologias ao longo do tempo. Isso motivou a criação do padrão *Open Archival Information System* (OAIS) [CCSDS 2002], que define as bases p na construção de repositórios para arquivar artefatos digitais a longo prazo, aptos a lidar com o impacto das mudanças. Associado ao OAIS está o *Metadata Encoding & Transmission Standard* (METS) [The Library of Congress 2004], um padrão de metadados para objetos digitais definido pela Biblioteca do Congresso dos Estados Unidos.

6. Considerações Finais

Este texto buscou delinear uma panorâmica considerando a problemática dos Objetos de Aprendizagem a partir de diversas perspectivas. Esperamos que ao final ele tenha contribuído para formar a noção de OA e suas características.

A área dos OAs tem contado com uma motivada e participativa comunidade de pesquisadores. Esperamos que este capítulo também contribua para esclarecer questões e desafios importantes nesse domínio.

Agradecimentos

Este trabalho foi parcialmente financiado pelos projetos LabMultflex (FAPESB-Infra 2006), MediaBank (HP Digital Publishing) e WebMaps II (CNPq).

Referências

ADL (2006). Sharable Content Object Reference Model (SCORM) 2004 – 3rd Edition – Overview. `http://www.adlnet.gov/downloads/DownloadPage.aspx?ID=237`, accessed on 10/2007.

Bachmann, F., Bass, L., Buhman, C., Comella-Dorda, S., Long, F., Robert, J., and Wallnau, R. S. K. (2000). Volume II: Technical Concepts of Component-Based Software Engineering, 2nd Edition. Technical Report CMU/SEI-2000-TR-008, Carnegie Mellon University.

Bekaert, J., De Kooning, E., and Van de Walle, R. (2005). Packaging models for the storage and distribution of complex digital objects in archival information systems: a review of MPEG-21 DID principles. *Multimedia Systems*, 10(4):286–301.

Broy, M., Deimel, A., Henn, J., Koskimies, K., Plášil, F., Pomberger, G., Pree, W., Stal, M., and Szyperski, C. (1998). What characterizes a (software) component? *Software – Concepts & Tools*, 19(1):49–56.

Burnett, I., Van de Walle, R., Hill, K., Bormans, J., and Pereira, F. (2003). MPEG-21: goals and achievements. *Multimedia*, 10:60–70.

CCSDS (2002). Reference Model for an Open Archival Information System (OAIS) – Blue Book. Technical Report CCSDS 650.0-B-1. `www.ccsds.org/CCSDS/documents/650x0b1.pdf`, accessed on 11/2004.

diSessa, A. A. (2001). The web/comp project – proposal. `http://www.soe.berkeley.edu/~boxer/webcomp/proposal.html`, accessed on 10/2007.

Dourado, P., Ferreira, P., and Santanchè, A. (2007). Representação Unificada de Objetos Digitais Complexos: confrontando o RAS com o IMS CP. In *Proc. of III Workshop on Digital Libraries*.

Ferreira, A. B. H. (2004). *Aurélio Século XXI, O Dicionário da Língua Portuguesa*. Editora Positivo.

Fox, E. A., Akscyn, R. M., Furuta, R. K., and Leggett, J. J. (1995). Digital libraries. *Commun. ACM*, 38(4):22–28.

Friesen, N., Fisher, S., and Roberts, A. (2004). Cancore guidelines – for the implementation of learning object metadata (ieee 1484.12.1-2002) version 2.0. `http://www.cancore.ca/en/guidelines.html`, accessed on 10/2007.

Houaiss, I. A. (2006). *Dicionário Houaiss da língua portuguesa*. Editora Objetiva.

IEEE L.T.S.C. (2000). Draft Standard for Learning Object Metadata – IEEE P1484.12/D4.0.

IEEE L.T.S.C. (2002). Draft Standard for Learning Object Metadata – IEEE 1484.12.1-2002. `http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf`, accessed on 09/2007.

L'Allier, J. J. (1997). Frame of Reference: NETg's Map to Its Products, Their Structures and Core Beliefs. `http://www.im.com.tr/framerefer.htm`, accessed on 10/2007.

McDonald, W. A., Hyde, J., and Montgomery, A. (2004). CMI Guidelines for Interoperability AICC. `www.aicc.org/docs/tech/cmi001v4.pdf`, accessed on 10/2007.

Mcilroy, M. D. (1968). Mass Produced Software Components. In Naur, P. and Randell, B., editors, *Software Engineering: Report of a conference sponsored by the NATO Science Committee*.

Nelson, M. L., Argue, B., Efron, M., Denn, S., and Pattuelli, M. C. (2001). A survey of complex object technologies for digital libraries – nasa/tm-2001-211426. Technical report.

Nilsson, M., Johnston, P., Naeve, A., and Powell, A. (2006). *The Future of Learning Object Metadata Interoperability*, pages 255–314. Informing Science Press.

Nilsson, M., Palmér, M., and Brase, J. (2003). The LOM RDF binding – principles and implementation. In *Proc. of the Third Annual ARIADNE conference*.

Olsen, G. (2006). From com to common. *Queue*, 4(5):20–26.

OMG (2005). Reusable Asset Specification – Version 2.2. `www.omg.org/docs/formal/05-11-02.pdf`, accessed on 08/2006.

Rockley, A. (2000). *Fundamental Concepts of Content Reuse*, chapter 2, pages 23–42. New Riders.

Roschelle, J., Digiano, C., Pea, R., and Kaput, J. (1998). Educational Software Components of Tomorrow (ESCOT). `http://www.escot.org/docs/MSET_ESCOT.html`, accessed on 10/2007.

Roschelle, J., DiGiano, C., Koutlis, M., Repenning, A., Phillips, J., Jackiw, N., and Suthers, D. (1999). Developing educational software components. *Computer*, 32(9):50–58.

Roschelle, J., Kaput, J., and Stroup, W. (2000). Simcalc: Accelerating students' engagement with the mathematics of change. *Educational technology and mathematics and science for the 21st century*, pages 47–75.

Santanchè, A. (2002). Explorando metáforas na elaboração de composições hipermídia. In *Anais do XXII Congresso da Sociedade Brasileira de Computação – VIII Workshop de Informática na Escola*, pages 389–396.

Santanchè, A. and Medeiros, C. B. (2007). A Component Model and Infrastructure for a Fluid Web. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):324–341.

Santanchè, A., Medeiros, C. B., and Pastorello Jr, G. Z. (2007). User-author centered multimedia building blocks. *Multimedia Systems*, 12(4):403–421.

Santanchè, A. and Teixeira, C. A. C. (2001). Anima: Promovendo Integração de Componentes na Web. In *Anais do VII Simpósio Brasileiro de Sistemas Multimídia e Hipermídia*, pages 261–268.

Santanchè, A. and Teixeira, C. A. C. (2002). Máquinas e componentes de software na aprendizagem matemática. In *Anais do XXII Congresso da Sociedade Brasileira de Computação – VIII Workshop de Informática na Escola*, pages 293–301.

Smythe, C. and Jackl, A. (2004). IMS Content Packaging Information Model – Version 1.1.4 Final Specification. Specification, IMS Global Learning Consortium, Inc. `www.imsglobal.org/content/packaging/cpv1p1p4/imscp_infov1p1p4.html`, accessed on 08/2007.

Sosteric, M. and Hesemeier, S. (2002). When is a learning object not an object: A first step towards a theory of learning objects. *The International Review of Research in Open and Distance Learning*, 3(2).

Szyperski, C. (2002). Component Software: Beyond Object-Oriented Programming. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

The Library of Congress (2004). METS: An Overview & Tutorial. `ww.loc.gov/standards/mets/METSOverview.v2.html`, accessed on 11/2004.

von Brevern, H. (2004). Cognitive and logical rationales for e-learning objects. *Journal of Educational Technology & Society*, 7(4):2–25.

Wiley, D. A. (2000). Connecting learning objects to instructional design theory: A definition, *a metaphor, and a taxonomy*. Online version: `http://reusability.org/read/chapters/wiley.doc`.